



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/729,773	12/08/2003	Jeffrey B. Scott	120 04984US	3422
128 7590 07/01/2009 HONEYWELL INTERNATIONAL INC. PATENT SERVICES 101 COLUMBIA ROAD P O BOX 2245 MORRISTOWN, NJ 07962-2245			EXAMINER WANG, JUE S	
			ART UNIT 2193	PAPER NUMBER
			MAIL DATE 07/01/2009	DELIVERY MODE PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

**Advisory Action
Before the Filing of an Appeal Brief**

Application No.

10/729,773

Applicant(s)

SCOTT, JEFFREY B.

Examiner

JUE S. WANG

Art Unit

2193

--The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

THE REPLY FILED 12 June 2009 FAILS TO PLACE THIS APPLICATION IN CONDITION FOR ALLOWANCE.

1. ☒ The reply was filed after a final rejection, but prior to or on the same day as filing a Notice of Appeal. To avoid abandonment of this application, applicant must timely file one of the following replies: (1) an amendment, affidavit, or other evidence, which places the application in condition for allowance; (2) a Notice of Appeal (with appeal fee) in compliance with 37 CFR 41.31; or (3) a Request for Continued Examination (RCE) in compliance with 37 CFR 1.114. The reply must be filed within one of the following time periods:

- a) ☐ The period for reply expires _____ months from the mailing date of the final rejection.
b) ☒ The period for reply expires on: (1) the mailing date of this Advisory Action, or (2) the date set forth in the final rejection, whichever is later. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of the final rejection.
Examiner Note: If box 1 is checked, check either box (a) or (b). ONLY CHECK BOX (b) WHEN THE FIRST REPLY WAS FILED WITHIN TWO MONTHS OF THE FINAL REJECTION. See MPEP 706.07(f).

Extensions of time may be obtained under 37 CFR 1.136(a). The date on which the petition under 37 CFR 1.136(a) and the appropriate extension fee have been filed is the date for purposes of determining the period of extension and the corresponding amount of the fee. The appropriate extension fee under 37 CFR 1.17(a) is calculated from: (1) the expiration date of the shortened statutory period for reply originally set in the final Office action; or (2) as set forth in (b) above, if checked. Any reply received by the Office later than three months after the mailing date of the final rejection, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

NOTICE OF APPEAL

2. ☐ The Notice of Appeal was filed on _____. A brief in compliance with 37 CFR 41.37 must be filed within two months of the date of filing the Notice of Appeal (37 CFR 41.37(a)), or any extension thereof (37 CFR 41.37(e)), to avoid dismissal of the appeal. Since a Notice of Appeal has been filed, any reply must be filed within the time period set forth in 37 CFR 41.37(a).

AMENDMENTS

3. ☐ The proposed amendment(s) filed after a final rejection, but prior to the date of filing a brief, will not be entered because
(a) ☐ They raise new issues that would require further consideration and/or search (see NOTE below);
(b) ☐ They raise the issue of new matter (see NOTE below);
(c) ☐ They are not deemed to place the application in better form for appeal by materially reducing or simplifying the issues for appeal; and/or
(d) ☐ They present additional claims without canceling a corresponding number of finally rejected claims.

NOTE: _____. (See 37 CFR 1.116 and 41.33(a)).

4. ☐ The amendments are not in compliance with 37 CFR 1.121. See attached Notice of Non-Compliant Amendment (PTOL-324).
5. ☐ Applicant's reply has overcome the following rejection(s): _____.
6. ☐ Newly proposed or amended claim(s) _____ would be allowable if submitted in a separate, timely filed amendment canceling the non-allowable claim(s).
7. ☒ For purposes of appeal, the proposed amendment(s): a) ☐ will not be entered, or b) ☒ will be entered and an explanation of how the new or amended claims would be rejected is provided below or appended.
The status of the claim(s) is (or will be) as follows:
Claim(s) allowed: NONE.
Claim(s) objected to: NONE.
Claim(s) rejected: 1-7 and 9-12.
Claim(s) withdrawn from consideration: NONE.

AFFIDAVIT OR OTHER EVIDENCE

8. ☐ The affidavit or other evidence filed after a final action, but before or on the date of filing a Notice of Appeal will not be entered because applicant failed to provide a showing of good and sufficient reasons why the affidavit or other evidence is necessary and was not earlier presented. See 37 CFR 1.116(e).
9. ☐ The affidavit or other evidence filed after the date of filing a Notice of Appeal, but prior to the date of filing a brief, will not be entered because the affidavit or other evidence failed to overcome all rejections under appeal and/or appellant fails to provide a showing a good and sufficient reasons why it is necessary and was not earlier presented. See 37 CFR 41.33(d)(1).
10. ☐ The affidavit or other evidence is entered. An explanation of the status of the claims after entry is below or attached.

REQUEST FOR RECONSIDERATION/OTHER

11. ☒ The request for reconsideration has been considered but does NOT place the application in condition for allowance because:
See Continuation Sheet.
12. ☐ Note the attached Information Disclosure Statement(s). (PTO/SB/08) Paper No(s). _____
13. ☐ Other: _____.

/Lewis A. Bullock, Jr./
Supervisory Patent Examiner, Art Unit 2193

Continuation of 11, does NOT place the application in condition for allowance because:

Applicant argued that Defaix does not teach the first validation step. Applicant argued "In paragraph 18 of the Office Action, the Examiner contends Defaix "not only teaches using files stored in the repository, but also teaches files that are not initially stored in the repository, i.e., 'some files will also already be stored in the sandbox', see 0032". This contention is mistaken. The client's sandbox contains only copies of versions of files stored in the repository. Paragraph 0032 clearly states that the sandbox 306 stores "local working copies of files from a corresponding project on the central server 100" and the "files in sandbox 306 are (possibly modified) particular versions of files from the repository 102". Paragraph 0032 further notes that the client "preferably does not have a local bulk data cache for the file contents" but can obtain them quickly via proxy server 200 as necessary". The statement that "some files will also already be stored in the sandbox 306" refers to the working copies set forth earlier in paragraph 0032. That is, some of the working copies of files in the repository may not contain the file contents, but the client can quickly obtain the file contents from proxy server 200. Therefore, the Examiner's contention is mistaken."

Applicant's arguments have been fully considered and Examiner respectfully disagrees. Examiner submits that contrary to Applicant's contention, Defaix does teach files that are not initially stored in the repository by teaching that "some files will also already be stored in the sandbox 306". While Defaix does state the files in the sandbox are particular versions of files from the repository, Defaix does not state that all files in the sandbox can only exist after it is first retrieved from the repository. Rather, a file that exists first in the sandbox but is later added to the repository is also a file that is a particular version of a file from the repository. The disclosure of files already stored in the sandbox appears after the disclosure that the client can obtain data from the proxy server as necessary. Therefore, Examiner interprets the phrase "Some files will also already be stored in the sandbox" to mean that these files does not need to be retrieved from the repository and are files that originated in the sandbox to be added to the repository. In addition, Applicant disagreed with Examiner's contention that developers are allowed to add new files to the repository over time because Defaix does not describe this feature. In response, Examiner submits while Defaix does not explicitly state that new files are added to the repository, original file versions do exist in the repository (i.e., version 1.1, see [0029]). It would have been obvious that Defaix supplies a means to add new original files to the repository since it is well known in the art that a software repository does not start with a fixed set of files, and techniques to add of new original files to a repository is well known in the art such as the CVS system taught in the cited prior art.

Applicant argued that Defaix does not teach the second validation step. Applicant stated "Paragraph 0007 located in the Background of the Invention section, describes a prior art locked checkout mechanism to control access to files. Defaix does not use this mechanism, but uses the mechanism described in paragraph 0038 that allows a first one of many clients working on a project to request a lock on a file that the first client is working on. In paragraph 19 of the Office Action, the Examiner argues "that checking if a file is locked is the same as checking if a file is checked in since a file is locked when a file is checked out (see 0007). Therefore, if a file is locked, then it is not checked in, and if a file is not locked, then it is checked in". This argument is mistaken. All of the existing files in repository 102 have a status of checked in. Defaix does not disclose or teach a checkout system that performs a lock at the time of checkout. Rather, Defaix distinguishes from the prior art system of paragraph 0007, by allowing multiple clients working on a project to access and possess working copies of the existing files in repository 102 that pertain to the project. So, in Defaix checkout is a meaningless event unless the existing file is locked. In order to lock an existing file, a client must request a lock. The procedure for granting the lock is shown in Fig. 5 and described in paragraph 0038. Notice that the fact that an existing file is locked by a first client prevents a second client from checking in a modified version of the existing file, but does not prevent the second client or another client from obtaining a copy of the existing file for its sandbox. For these reasons, there is no need for Defaix to determine "if said existing object has a status of checked in" because all of the existing files are checked in. Therefore, Defaix lacks the second step."

Applicant's arguments have been fully considered and Examiner respectfully disagrees. Examiner submits that while the locked checkout mechanism is disclosed in paragraph [0007] that appears the background section, it does not imply that Defaix does not use this locked checkout mechanism. The novelty of Defaix is in providing proxies with read-only caches to provide fast and reliable access to the data (see [0011], [000051], [0052]). The use of proxies does not exclude Defaix from using a technique that is employed by existing software repository management systems such as locked checkout. In fact, Defaix teaches "a checkout mechanism for controlling modification to the repository" (see page 5, claim 7). The fact that Defaix teaches a lock that is requestable by a user does not exclude the use of a locked checkout mechanism which is also taught in the background section of Defaix. Defaix teaches that multiple clients are allowed to work on a project to access and possess working copies of the existing files in the repository as stated by the Applicants, but these working copies do not affect the repository until it is checked into the repository and modification to the data in the repository is controlled by the checkout mechanism and the lock mechanism.

Applicant argued that Defaix does not teach the third step of the validation function because there is no teaching or disclosure in Defaix that the access control list is used at the time of check in of a modified version of an existing file.

Applicant's arguments have been fully considered and Examiner respectfully disagrees. Defaix teaches that the access control list is used to control who has access to objects in the repository (see [0039]) and manage requests from clients to modify the repository (see claim 15, part c). A check in of a new version is a request to modify the repository, therefore, Examiner submits that access control is used by the validation function described in paragraph [0035].

Applicant argued that Defaix does not teach the fourth step of the validation function. Applicant stated "The prior art locked checkout mechanism disclosed by Defaix in paragraph (0007) locks an object upon checkout and not upon check-in as recited in claimed fourth step of the validation function. There is no teaching in Defaix to modify the locked checkout mechanism of paragraph 0007 to be a locked check-in mechanism or to even use the paragraph 0007 procedure. Therefore, the Examiner's contention and conclusion are erroneous. ... In fact, the only teaching in Defaix for a lock procedure is shown in Fig. 5 and described in paragraph 0038 for locking a file. This procedure is independent of check in or checkout and must be separately requested by the client."

Applicant's arguments have been fully considered and Examiner respectfully disagrees. In response, Examiner submits that Defaix teaches "only the developer who owns the lock can modify the file by checking in a new version". (see [0007]) and a mechanism to request a lock (see [0038]). While Defaix does not explicitly teach requesting the lock at check in, Defaix teaches a lock is required for check in. In the system of Defaix where multiple clients have access to the files via the read-only cache on proxies, an explicit checkout might not have been executed and therefore, no lock yet exist on the file being checked in. Therefore, in such situations, a lock can be requested for the check in to prevent modification conflicts. Defaix does not teach that the locking is performed automatically. However, the automating of manual activities which accomplishes the same result is not sufficient to distinguish over the prior art (see MPEP, 2144.04 [R6] III. Automating a Manual Activity)..